requesting the encapsulation object containing the platform dependent method from the system manager by the platform independent object;

retrieving the business card corresponding to the requesting object by the system manager;

retrieving the encapsulation object pointer from the requesting object's business card by the system manager;

instantiating the encapsulation object identified by the encapsulation object pointer;

calling a platform independent wrapper contained in the encapsulation object corresponding to the platform dependent method by the platform independent object; and calling the platform dependent method by the wrapper.

REMARKS

A number of claims were rejected under U.S.C. 102 as being anticipated by "the Java Platform A White Paper" by Kramer providing an overview of the Java Platform and its various components. In particular, the Examiner cites Fig. 2 and the accompanying narrative at page 17 that describes, "the Java Base Platform in part shown in black, including the blocks labeled Adapter. The Java API includes both the Java Core API and the Java Standard Extension API...The Porting Interface lies between the Java Virtual Machine and the operating system (OS) or browser. The Porting Interface has a platform independent part (shown in black) and a ***platform dependent part, shown as Adapters***." In support of his 102 rejection of claim 1, the Examiner equates the claimed wrapper with the adapter (i.e., "a wrapper(adapter) arranged to call the platform dependent method"). Based upon the cited reference, therefore, the adapter is platform dependent such that each OS or browser has an associated adapter that is used to interface the particular OS and the Porting Interface (see Fig. 2 of Kramer) which, in turn, interfaces with the Java Virtual Machine (JVM).

In contrast, claim 1 provides,

> "a platform independent wrapper arranged to call the platform dependent method, wherein a platform independent object accesses the platform dependent method by calling the wrapper, wherein the wrapper then calls the platform dependent method".

Therefore, the wrapper recited in claim 1 is a platform independent wrapper that is used to call the platform dependent method. In this way, a particular wrapper can be used to call any one of a number of different methods each of a different platform, if so desired. Support is found at page 5 line 26 – page 6 line 1, "the Java wrapper 102a (which is be definition platform independent)...".

Accordingly, the Applicant believes that claim 1 is neither anticipated nor reasonably suggested by the cited reference and the Examiner is respectfully request to withdraw his 102 rejection of claim 1.

The Examiner rejected a number of claims (particularly independent claim 17 and associated dependent claims 18 – 20) under 35 U.S.C. 103(a) over Kramer in view of U.S. Patent 6,134,616 issued to Beatty describing a business card that includes configuration data that adds nothing, however, to the primary reference to render claim 17 as being obvious. In particular, claim 17 recites,

> "calling a platform independent wrapper contained in the encapsulation object corresponding to the platform dependent method by the platform independent object; and
> calling the platform dependent method by the wrapper."

As with claim 1 above, neither reference taken singly or in any combination anticipates or reasonably suggests the invention as recited in claim 17 which the Applicant believers to be allowable over the cited art for at least the same reasons as claim 1

All remaining dependent claims depend either directly or indirectly on independent claims 1, 10 and 17 and are therefore also allowable over the cited art for at least the reasons stated for claims 1, 10 and 17.

CONCLUSION

In view of the foregoing, it is respectfully submitted that all pending claims are allowable. Should the Examiner believe that a further telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

[illegible]

Michael J. Ferrazano
Reg. No. 44,105

P.O. Box 778
Berkeley, CA 94704-0778
(650) 961-8300

MARKED UP VERSION OF AMENDED CLAIMS

1. (Once Amended) A software object included in a computer system, comprising:

a platform dependent method; and

a <u>platform independent</u> wrapper arranged to call the platform dependent method, wherein a platform independent object accesses the platform dependent method by calling the wrapper, wherein the wrapper then calls the platform dependent method.

10. (Once Amended) A computer-implemented method of accessing a platform dependent method by a platform independent object in a computer system, the computer system having an encapsulation object [that includes a wrapper associated with the method], the method comprising:

calling [the]<u>a platform independent</u> wrapper by the platform independent object; and

calling the method by the [first] wrapper <u>wherein the wrapper is included in the encapsulation object</u>.

17. (Once Amended) A computer-implemented method of accessing a platform dependent method by a platform independent object in a computer system, the computer system including a system manager, an encapsulation object that contains the platform dependent method [and a wrapper that is used to call the platform dependent method,] and a business card associated with the platform independent object, the business card containing configuration data that includes an encapsulation object pointer that is used to identify the encapsulation object containing the platform dependent method, comprising:

requesting the encapsulation object containing the platform dependent method from the system manager by the platform independent object;

retrieving the business card corresponding to the requesting object by the system manager;

retrieving the encapsulation object pointer from the requesting object's business card by the system manager;

instantiating the encapsulation object identified by the encapsulation object pointer;

calling [the]a platform independent wrapper contained in the encapsulation object corresponding to the platform dependent method by the platform independent object; and calling the platform dependent method by the wrapper.

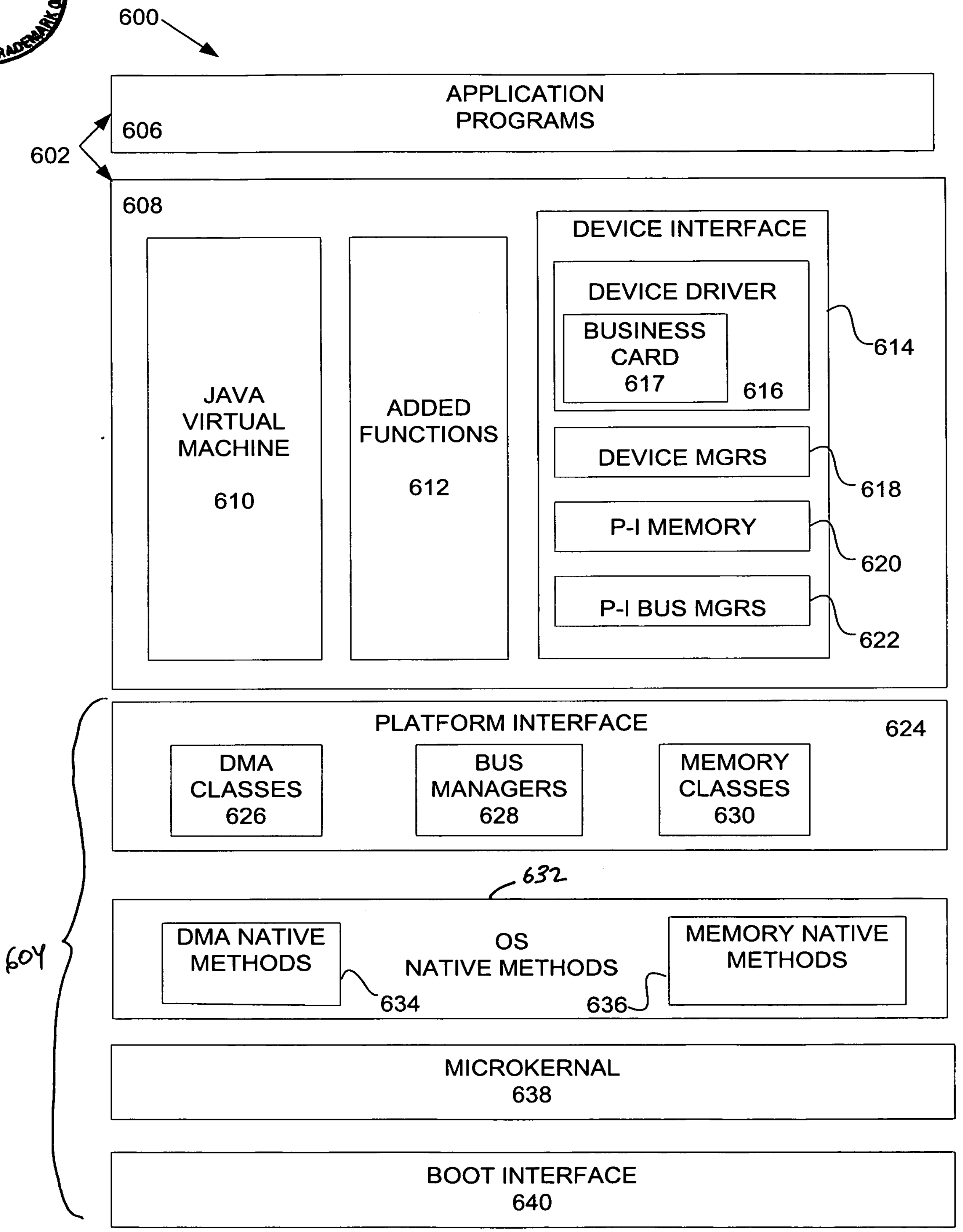
600
602
APPLICATION PROGRAMS
606
608
JAVA VIRTUAL MACHINE 610
ADDED FUNCTIONS 612
DEVICE INTERFACE
DEVICE DRIVER
BUSINESS CARD 617
616
614
DEVICE MGRS
618
P-I MEMORY
620
P-I BUS MGRS
622
PLATFORM INTERFACE
624
DMA CLASSES 626
BUS MANAGERS 628
MEMORY CLASSES 630
632
DMA NATIVE METHODS
OS NATIVE METHODS
MEMORY NATIVE METHODS
634
636
604
MICROKERNAL 638
BOOT INTERFACE 640

FIGURE 6